Taking into account the execution context to optimize medical image databases indexation

Diane Lingrand, Tristan Glatard, Johan Montagnat

Taking into account the execution context to optimize medical image databases indexation.

Diane Lingrand[1], Tristan Glatard[12], Johan Montagnat[1]

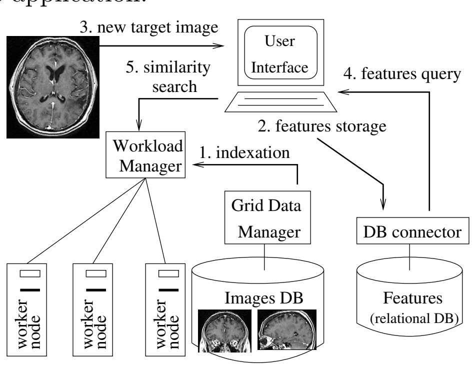
[1] I3S laboratory - University of Nice - Sophia Antipolis and CNRS

http://www.i3s.unice.fr/~lingrand

[2] INRIA Sophia, Asclepios, http://www-sop.inria.fr/asclepios

1. Indexing image databases

Grid infrastructures have been successfully exploited for production in the area of medical image analysis for several years. Grids enable the emergence of new applications involving large amounts of data such as epidemiology, statistical studies or medical image databases search. In [?], we proposed the deployment of a medical image database indexation and retrieval procedure on a grid infrastructure. The computational complexity comes from two steps: (i) the indexation of all images in the database and (ii) the search for images based on a similarity criterion. The following figure roughly depicts the application.



Images from the database are first distributed to grid worker nodes for concurrent analysis (1). Search features (*e.g.* histograms, texture parameters...) are extracted and stored in a database (2). When a user brings a new image into the system and searches for similar cases (3), a query is made to the database to identify a collection of close candidate images (4). The candidate images are then compared to the target concurrently on the grid nodes (5).

This application exhibits a massive parallelism in both steps (1) and (5): they can easily be distributed. However, their grid enactment is non trivial for two reasons: (i) performance drops due to the high latency encountered on production grids and (ii) system faults making the trivial parallelization unreliable. Indeed production grids are characterized by permanent but non-stationary load and a large geographical extension. As a consequence, latency, measured as the time between the submission time of a computation job and the beginning of its execution, can be very high and experience large variations. As an example, on the EGEE production grid[1] the average latency for each computing task is in the order of 5 minutes with standard deviation also in the order of 5 minutes. The long latency penalizes the application efficiency and the high variability make it difficult to optimize. In addition, a non negligible ratio of faults ($\rho \sim 2.5\%$) cause some of the computation tasks to never end.

2. Grid Modeling

To improve the performance and reliability of this application, it is necessary to accurately evaluate the computing tasks execution time. Abnormally long tasks (due to faults and long variations) can then be time-outed and resubmitted. In [2], a probabilistic model of the execution time function of the timeout value t_∞, the proportion of faults ρ and of the latencies distribution f_R is proposed (where R is a random variable). Let F_R be the cumulative density function (cdf). The optimal timeout value can be obtained by minimizing the expectation of the

[1]EGEE, http://www.eu-egee.org/

job execution time J, $E_J(t_\infty) =$

$$\frac{1}{F_R(t_\infty)}\int_0^{t_\infty} u f_R(u)du + \frac{t_\infty}{(1-\rho)F_R(t_\infty)} - t_\infty \quad (1)$$

Some parameters from the execution context have an influence on the cumulative density function of latencies [1]. In this paper, we quantify their influence on the timeout values and the expected execution time (including resubmissions). We aim at refining our model by taking into account most relevant contextual parameters.

3. Results

Experiments on EGEE. On the EGEE grid, user jobs are sent to a Resource Broker (RB) which dispatches them to the different computing centers available. The gateway to each computing center is one or more Computing Element (CE). A CE hosts a batch manager that will distribute the workload over the center worker nodes, using different CE-queues. Different queues are handling jobs with different wall clock time.

Execution context. Each job can be characterized by its execution context that depends on the grid status and may evolve during its life-cycle. The context of a job depends both on parameters internal and external to the grid infrastructure. The internal context corresponds to parameters such as the computer(s) involved in the WMS of a specific job. It may not be completely known at the job submission time. The external context is related to parameters such as the day of the week or the submission time that may have an impact on the load imposed to the grid. Below we study the impact of the target CE (variable computing sites performance and variable load conditions) and the correlation with the time of the day (working and week-end days are expected to be correlated to different system loads).

Influence of the computing center (CE). We have observed that a majority of the 92 CE-queues available on the EGEE grid exhibit similar cdfs while others are more singular. A k-mean classification algorithm was used to split them into two groups. For each cdf the optimal timeout value is computed, by minimizing equation 1. The ANOVA statistical test shows that the means of timeout values in each set differ significantly ($F = 0.0003$, $p < 0.1\%$).

Influence of the days of the week combined with RBs. The day of the week can potentially influence the optimal timeout value since the activity is likely to be maximal during the week days (thus increasing the system load) and minimal during week-ends. Conversely, system maintenance is expected to be better during week days than during week-ends.

In this experiment, the day of the week information is correlated with the Resource Broker to evaluate the mutual influence of both parameters. The minimum optimal timeout value observed is 436 seconds while the maximum is 1105 seconds. The difference is about 11 minutes. Taking into account both RBs and day can thus drastically improve the total job execution time.

4. Conclusions, references and acknowledgments

Grid are promising tools to tackle massively parallel medical applications such as medical image databases indexation but their complexity make the optimization of computation tasks and the design of computing models difficult. We tackle this complexity by a probabilistic approach. We have shown that a probabilistic model of the computing tasks execution time can be refined by taking into account the execution context which varies over the grid infrastructure. This modeling and the resulting optimization strategy enable the deployment of reliable and efficient large scale applications.

[1] J. Montagnat, V. Breton and I. Magnin Partitioning medical image databases for content-based queries on a grid. *Methods of Information in Medicine*, 44(2):154–160, 2005.

[2] T. Glatard, D. Lingrand, J. Montagnat, and M. Riveill. Impact of the execution context on Grid job performances. In *WCAMG'07*, Rio de Janeiro, May 2007. IEEE.

[3] T. Glatard, J. Montagnat, and X. Pennec. Optimizing jobs timeouts on clusters and production grids. In *CCGrid*, Rio de Janeiro, May 2007. IEEE.

This work is partially funded by the NeuroLOG ANR-06-TLOG-024 project and the ICT-Asia OnCoMedia program.